# Light Field Warping

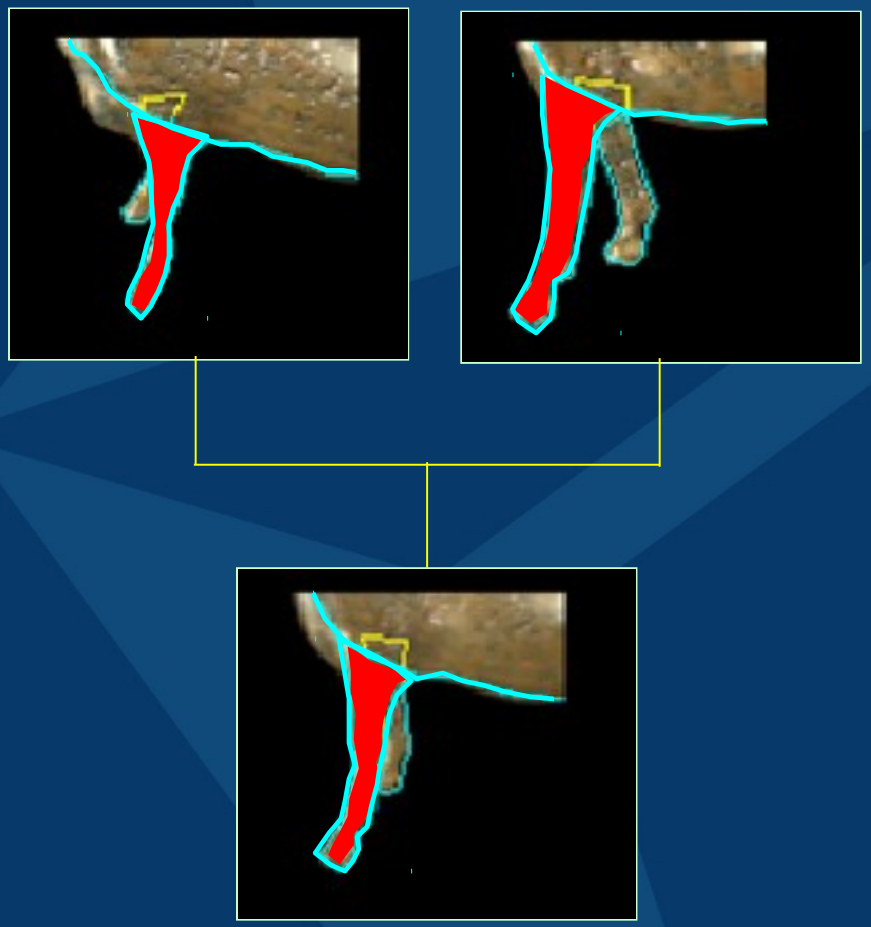
**Given $L_0$ & $L_1$, warp $L_0$ to $L'$**

- Obtain feature elements of $L'$
- Compute GVM of $L'$
- Warp ray bundles of $L_0$ view-by-view
  - Ray-space warping equation
- Treat background rays (pixels)

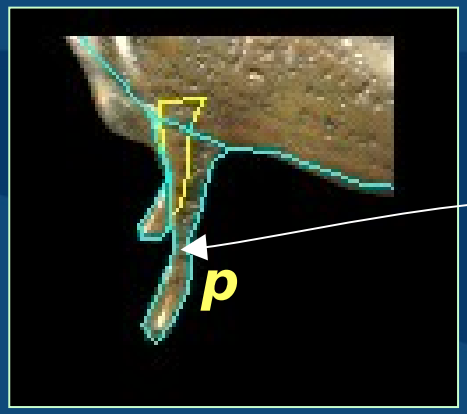
# Feature Elements of *L'*

**Linearly interpolated from *L0* & *L1***

- Feature points, lines, & polygons
- Background edges



# Warping Ray Bundles




$L_0$ (8,0)

**Use the ray-space warping equation to find**

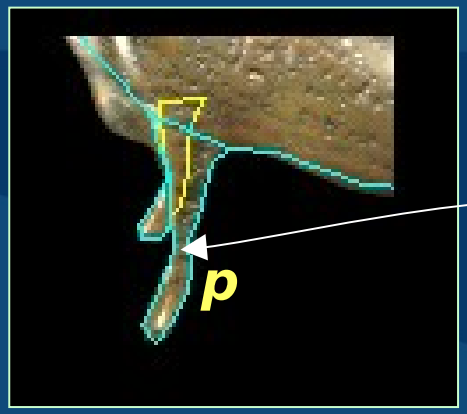
***p*** **= pre-image (*p'*)**

**in the same view of *$L_0$***

***p'***

$L'$ (8,0)

# Warping Ray Bundles




$L_0$ (8,0)

If ***p*** visible

color (***p'***) = color (***p***)

else

***p'*** is in a hole

***p'***

$L'$ (8,0)

# Ray-space Warping

- **Motivation: Fill the holes encountered during ray bundle warping**
  - Holes are caused by visibility changes (due to object shape changes)
  - Holes can be arbitrarily large
- **Basic idea: Approximate occluded rays by taking rays from nearby views**

# Ray-space Warping

$L_0$ (8,0)

***p*** = pre-image (***p'***) not visible in the same view of ***Lo***

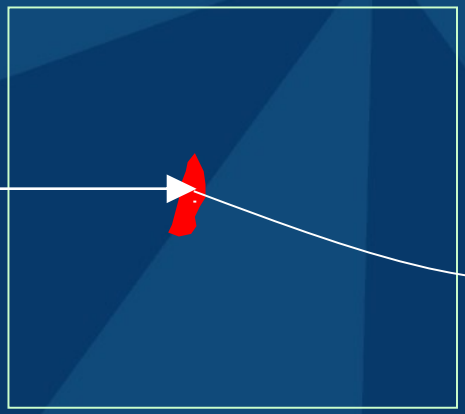
***p'***

$L'$ (8,0)

# Ray-space Warping

$L_0$ (8,0)

$L_0$ (24,8)



***p'***

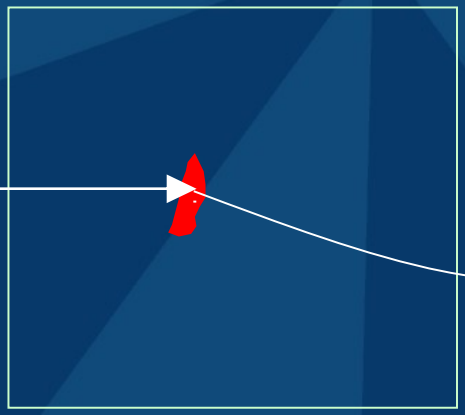
**pre-image(*p'*) visible in a different view of *L0***

$L'$ (8,0)

# Ray-space Warping

$L_0$ (8,0)

$L_0$ (24,8)

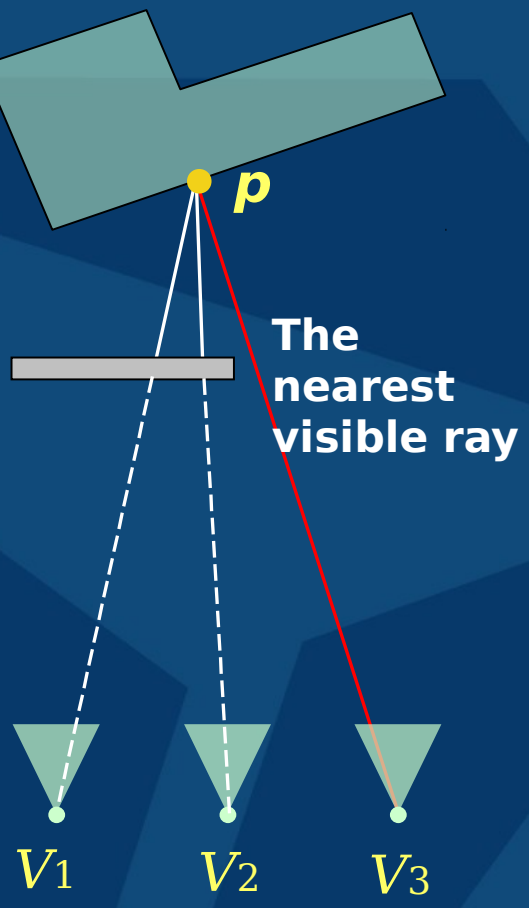


***p'***

$L'$ (8,0)

**color (*p'*) = color (*r*), where *r* = nearest visible ray of *p***

# Nearest Visible Ray




- **A heuristic to best approximate non-Lambertian surfaces**
- **Search for nearest visible rays**
  - Starts from $V_1$'s immediate neighbors
  - Expands out in breadth-first order
- **Search will never fail**

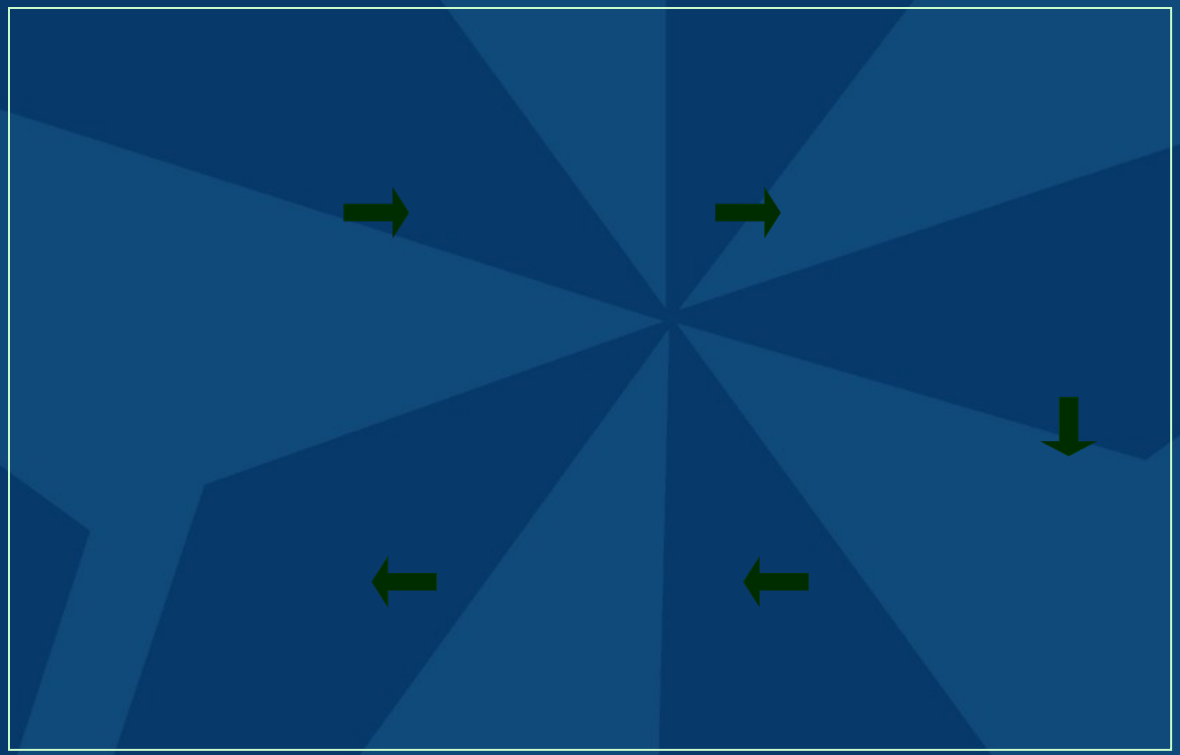
# 3D Facial Morphing

# Complex Surface Properties

# Large Visibility Change

# Other Applications

## Key-frame morphing



# Other Applications

## Plenoptic texture transfer

# Other Applications

# Discussion

- **LF morphing generalizes image morphing**
- **Compared w. geometry morphing**
  - Handling complex surface properties is easy with LF morphing
  - LF morphs are only good for viewing (& for restricted viewing range only)

# Conclusion

- **A general framework for image-based 3D morphing**
  - Based on ray correspondence
  - Easy-to-use UI for specifying features
  - Ray-space warping for visibility changes
- **Future topics: topology change, more tasks automated, other LF operations.**

# Acknowledgements


- Hua Zhong (initial LFM prototype)
- Steve Lin & Yin Li (video production)
- Xin Tong & Sing Bing Kang